Amendments to the Claims

The listing of claims will replace all prior versions, and listings of claims in the application.

1. (Currently Amended) A context-selection mechanism within a microprocessor for selecting a context from a pool of contexts for processing a data packet having a header, comprising:

an interface for receiving [[a]] the data packet and communicating with a multi-streaming processor, said multi-streaming processor comprising a core that includes the pool of contexts;

circuitry for computing input data into a value according to one or more logic rules and for selecting a context from the pool of contexts based at least in part on the value; and

a loading mechanism for preloading data corresponding to the received data packet into the selected context for subsequent processing, the preloaded data including information from the data packet header;

wherein computation of the input data functions to enable identification and selection of a context for packet processing according to the logic rule at the instant time such that a multitude of context selections made over a period of time facilitate balancing of load pressure on functional units housed within the multi-streaming processor core and required for packet processing.

2. (Original) The context-selection mechanism of claim 1 integrated to a data packet router operating in a data-packet-network.

3. (Original) The context-selection mechanism of claim 2 wherein the data-packet-network is the Internet network.

4. (Previously Presented) The context-selection mechanism of claim 1 wherein the pool of contexts is divided into separate clusters in the core, each cluster containing some of the functional units used in packet processing.

5. (Original) The context-selection mechanism of claim 1 wherein the input data into the computation circuitry includes availability information of individual ones of the pool of contexts at the time of computation.

6. (Original) The context-selection mechanism of claim 5 wherein the input data into the computation circuitry further includes real time information of any processing streams stalled in un-available ones of the pool of contexts and the reason for the stall.

7. (Original) The context-selection mechanism of claim 5 wherein the input data into the computation circuitry further includes statistical data about previous processing time periods required to process similar data packets.

8. (Original) The context-selection mechanism of claim 5 herein the input data into the computation circuitry further includes statistical data about the distribution of

instruction types associated with individual ones of previously processed and similar data packets.

9. (Original) The context-selection mechanism of claim 1 wherein the input data is sourced from the multi-streaming processor.

10. (Original) The context-selection mechanism of claim 1 wherein the input data is sourced from a third party.

11. (Original) The context-selection mechanism of claim 4 wherein the clusters are numbered and the functional units are distributed symmetrically therein.

12. (Original) The context-selection mechanism of claim 4 wherein the clusters are numbered and the functional units are distributed asymmetrically therein.

13. (Currently Amended) A system for load balancing pressure on functional units within a multi-streaming processor core during the processing of multiple data packets comprising:

a context-selection mechanism having a communication interface;

circuitry for computing input data into a value according to one or more logic rules and a mechanism for preloading data corresponding to a data packet received from the communication interface into available ones of a pool of contexts within the core, the preloaded data including at least a portion of a data packet header of the data packet;

a multi-streaming processor core responsible for processing the data packets, the processor core hosting the functional units and the context pool; and

a set of instructions comprising the one or more logic rules governing context selection, wherein packet processing pressure upon the functional units within the processor core is balanced by selecting individual contexts for processing packet information based at least in part on the value.

14. (Previously Presented) The system of claim 13 integrated to a data packet router operating in a data-packet-network.

15. (Previously Presented) The system of claim 14 wherein the data-packet-network is the Internet network.

16. (Previously Presented) The system of claim 13 wherein the pool of contexts is divided into separate clusters in the processor core, each cluster containing some of the functional units used in packet processing.

17. (Previously Presented) The system of claim 13 wherein the input data into the computation circuitry includes availability information of individual ones of the pool of contexts at the time of computation.

18. (Original) The system of claim 13 wherein the input data into the computation circuitry further includes real time information of any processing streams stalled in un-available ones of the pool of contexts and the reason for the stall.

19. (Original) The system of claim 13 wherein the input data into the computation circuitry further includes statistical data about previous processing time periods required to process similar data packets.

20. (Original) The system of claim 13 wherein the input data into the computation circuitry further includes statistical data about the distribution of instruction types associated with individual ones of previously processed and similar data packets.

21. (Original) The system of claim 13 wherein the input data is sourced from the multi-streaming processor and provided in a software table.

22. (Original) The system of claim 13 wherein the input data is sourced from a third party.

23. (Original) The system of claim 16 wherein the clusters are numbered and the functional units are distributed symmetrically therein.

24. (Original) The system of claim 16 wherein the clusters are numbered and the functional units are distributed asymmetrically therein.

25. (Original) The system of claim 13 wherein the set of instructions comprising the logic rule is programmable.

26. (Currently Amended) A method for load balancing pressure on functional units contained within a multi-streaming processor core during processing of multiple data packets comprising steps of:

(a) arranging the functional units into two or more separate clusters on the core of the processor, each of said clusters containing an equal number of contexts that may write to functional units which are hosted by a corresponding cluster, wherein said functional units and contexts are included in the processor core;

(b) receiving a data packet for processing, the data packet having a header;

(c) receiving as input for computation, data about the instant availability status of individual contexts within each cluster;

(d) receiving as input for computation, data about stream status of streams occupying any contexts within each cluster; and

(e) computing the data received as input to produce a value, the value identifying and initiating selection of a context for processing packet information and balancing packet processing load of the functional units within each cluster;

(f) preloading data corresponding to the received data packet into the selected context for subsequent processing, the preloaded data including information from the data packet header; and

(g) repeating steps (b) through (f) for each of the multiple data packets for processing.

27. (Original) The method of claim 26 practiced in conjunction with a data packet router operating in a data-packet-network.

28. (Original) The method of claim 27 wherein the data-packet-network is the Internet network.

29. (Original) The method of claim 26 wherein in step (a) the functional units are provided within each cluster in a symmetrical fashion.

30. (Original) The method of claim 26 wherein in step (a) the functional units are provided within each cluster in an asymmetrical fashion.

31. (Original) The method of claim 26 wherein in step (b) the packet is received at a data port of a data router and requires automatic activation.

32. (Original) The method of claim 26 wherein in step (b) the packet is held by the processor and requires a context for processing.

33. (Previously Presented) The method of claim 26 wherein in step (c) availability status comprises an indication of which one of two components owns each context.

34. (Original) The method of claim 33 wherein in step (c) one of the components is the processor and other component is a packet management unit.

35. (Original) The method of claim 26 wherein in step (d) the data about stream status includes whether or not streams are stalled within any of the contexts and the reason for each instance of a stalled stream.

36. (Original) The method of claim 26 wherein in step (d) the data about stream status includes time parameters of how long each stream will take to process data packets associated with their contexts.

37. (Original) The method of claim 26 wherein in step (d) the data about stream status includes distribution parameters of instruction types that each stream has executed to process its data packet.

38. (Original) The method of claim 26 wherein in steps (c) through (d) are practice according to the rule of logic.

39. (Previously Presented) The method of claim 26 wherein the rule of logic is programmable.

40. (Previously Presented) The context selection mechanism of claim 1, wherein the data corresponding to the received data packet includes:

an address indicating the start of a memory region in which the received data packet is stored; and

at least a portion of a header of the received data packet.

41. (Previously Presented) The context selection mechanism of claim 40, wherein the loading mechanism is configured to:

convey a first indication to the multi-streaming processor at the start of preloading data corresponding to the received data packet; and

convey a second indication to the multi-streaming processor at the completion of preloading data corresponding to the received packet.

42. (Previously Presented) The system of claim 13, wherein the data corresponding to the received data packet includes:

an address indicating the start of a memory region in which the received data packet is stored; and

at least a portion of a header of the received data packet.

43. (Previously Presented) The system of claim 42, wherein the mechanism for preloading data is configured to:

convey a first indication to the multi-streaming processor at the start of preloading data corresponding to the received data packet; and

convey a second indication to the multi-streaming processor at the completion of preloading data corresponding to the received packet.

44. (Previously Presented) The method of claim 26, wherein the data corresponding to the received data packet includes:

an address indicating the start of a memory region in which the received data packet is stored; and

at least a portion of a header of the received data packet.

45. (Previously Presented) The method of claim 44, further comprising:

conveying a first indication to the multi-streaming processor at the start of preloading data corresponding to the received data packet; and

conveying a second indication to the multi-streaming processor at the completion of preloading data corresponding to the received packet.